

DEI
determining based on the detected event whether an animated sequence does not reflect the state of the container; and
updating the cyclical display based on the determination.--

REMARKS

In the Office Action, the Examiner rejected claims 1, 13, 29, and 31 under 35 U.S.C. §103(a) as being unpatentable over Baecker et al. (U.S. Patent No. 5,479,602) in view of Johnston, Jr. et al. (U.S. Patent No. 5,880,729) (Office Action [hereinafter "OA"], paragraph 3); rejected claims 2-4, 10, 14 and 16 under 35 U.S.C. § 103(a) as being unpatentable over Baecker et al. and Johnston, Jr. et al., and further in view of Gudmundson et al. (U.S. Patent No. 5,680,619) (OA, paragraph 4); rejected claims 5-6, 9, 11-12, 15, 17 and 18 under 35 U.S.C. § 103(a) as being unpatentable over Baecker et al., Johnston, Jr. et al., and Gudmundson, and further in view of Gallagher (Computer Visualization: Graphics Techniques for Scientific and Engineering Analysis) (OA, paragraph 5); rejected claims 7, 19-21, 25 and 30 under 35 U.S.C. § 103(a) as being unpatentable over Johnston, Jr. et al. (OA, paragraph 6); rejected claims 22-24 and 26 under 35 U.S.C. § 103(a) as being unpatentable over STN Express, Lagarde et al. (U.S. Patent No. 5,721,908), and Johnston, Jr. et al. (OA, paragraph 7); rejected claims 27 and 28 under 35 U.S.C. § 103(a) as being unpatentable over Baecker et al., STN Express, Lagarde et al., and Johnston, Jr. et al. (OA, paragraph 8); and rejected claim 8 under 35 U.S.C. § 103(a) as being unpatentable over Johnston, Jr. et al. in view of Gallagher (OA, paragraph 9).

By this Amendment, Applicants add new claim 32 to further define certain aspects of the present invention. Accordingly, in view of the remarks that follow, Applicants respectfully traverse the Examiner's rejections of claims 1-31 under 35 U.S.C. § 103(a).

The Examiner rejected claims 1, 13, 29, and 31 under 35 U.S.C. § 103(a) as being unpatentable over Baecker et al. in view of Johnston, Jr. et al. (OA, paragraph 3). The Examiner alleged that Baecker et al. discloses, *inter alia*, the modification of an animation icon when a corresponding file or folder changes, so that new animation frames are generated. The Examiner admitted that Baecker et al. fails to teach transitional visual effects or animation for allowing users to view the transition of an object between two different static states of elements and relied on Johnston, Jr. et al. to teach that feature. The Examiner also alleged that Johnston, Jr. et al. discloses detecting an event reflecting a change in the state of the container. The Examiner alleged that it would have been obvious to determine based on a detected event whether an animated sequence does not reflect the state of the container because doing so allows the user to detect whether a change has occurred to a computer file while the user passively observes without entering additional inputs.

The present invention recited in claim 1, for example, recites a process for reflecting a state of a software container having objects, comprising: cyclically displaying a series of frames reflecting a state of the container as an animated sequence; detecting an event reflecting a change in the state of the container; determining based on the detected event whether an animated sequence does not reflect the state of the container; and updating the cyclical display based on the determination.

Applicants respectfully submit that Baecker et al. and Johnston, Jr. et al. do not disclose or suggest this claimed combination of steps. Among other things, the references do not disclose or suggest the step of determining based on the detected event whether an animated sequence does not reflect the state of the container.

Baecker et al. discloses an apparatus and method for generating and displaying a content-based animated depiction of a standard icon. The animated depiction can reflect the state of a folder that has several files in it and can change the animation scenario whenever the folder represented by the animated icon is modified (col. 8, line 58 - col. 9, line 6). In contrast, the animated sequence according to the present invention is changed when it has been determined that the animated sequence does not reflect the state of a container. In this manner, unnecessary update cycles can be eliminated thereby improving performance. This is reflected in the written description with regard to an implementation of the present invention. Upon the detection of the event that reflects a change in state, animated indicator program 206, through monitor program 208, determines whether the state of the software container has changed enough to warrant a revised animation cycle (page 7, lines 1-3). If there has not been enough of a change, then no revised animation cycle is needed. As a consequence of this configuration, the system does not even attempt to update the cyclical display when an update is not necessary.

In Baecker et al., updates occur every time there is a folder modification, thus wasting valuable time if it turns out that the animation scenario does not have to be changed at all. In other words, Baecker et al. keeps its cyclical display accurate by regenerating the frames that constitute the cyclical display every time a modification to the folder occurs. There is no quality control as to whether the regeneration is needed. For example, if a user of the system in Baecker et al. were to delete one file and create another file in between update cycles, the system would detect that a folder modification has occurred and cause a regeneration of the frames. In this case, however, a regeneration was not needed because the number of files in the folder remained the same. Thus, the system performed a regeneration that was not necessary to ensure that the cyclical display reflected the state of the folder. In contrast, a system consistent with the present

invention eliminates that unnecessary regeneration by determining whether the animated sequence reflects the state of the folder. Accordingly, Baecker et al. does not disclose or suggest the step of determining based on the detected event whether an animated sequence does not reflect the state of the container.

The teachings of Johnston, Jr. et al. are not sufficient to make up for the deficiencies of Baecker et al. As stated by the Examiner, Johnston, Jr. et al. discloses the use of animation for allowing users to view the transition of an object between two different static states of elements within a GUI. There is nothing in Johnston, Jr. et al., however, to suggest determining based on the detected event whether an animated sequence does not reflect the state of the container.

The Examiner asserted that it would have been obvious to determine based on a detected event whether an animated sequence does not reflect the state of the container because doing so allows the user to detect whether a change has occurred to a computer file while the user passively observes without entering additional inputs. Applicants respectfully disagree with that position. As explained before, neither Baecker et al. nor Johnston, Jr. et al. disclose the aforementioned determining step. Furthermore, Applicants submit that animation only while the state is being changed is not sufficient to teach determining whether an animated sequence does not reflect the state of the container. Animating only while the state is being changed might be evidence that an update of the animated sequence occurs, but does not prove that a separate determination occurs. The present invention includes both a determination and an update. Accordingly, neither Baecker et al. nor Johnston, Jr. et al. disclose or suggest the step of determining based on the detected event whether an animated sequence does not reflect the state of the container. The combination of Baecker et al. and Johnston, Jr. et al. also fails to teach this claim element.

LAW OFFICES

FINNEGAN, HENDERSON,
FARABOW, GARRETT,
& DUNNER, L.L.P.
1300 I STREET, N. W.
WASHINGTON, DC 20005
202-408-4000

Moreover, Baecker et al. actually teaches away from such a determination. Baecker et al. discloses that when an object is modified, the associated animation can be regenerated on a periodic basis (col.8, lines 63-65). This regeneration on a periodic basis is essentially an unconditional update of the frames that constitute the animation. Periodic regeneration is unconditional because the regeneration occurs every given time period without checking other factors. On the other hand, systems consistent with the present invention, utilize an update function that occurs conditionally based on whether the current animation sufficiently reflects the state of the container. It is not obvious to change an unconditional update, such as that taught by Baecker et al., to a conditional update, such as that taught by the present invention. Furthermore, there is no teaching or suggestion in either Baecker et al. or Johnston, Jr. et al. to alter the combined system of the references so that there is an update conditional on an animated sequence matching the container state. Accordingly, it would not have been obvious to determine based on a detected event whether an animated sequence does not reflect the state of the container.

For at least the foregoing reasons, Applicants submit that claim 1 is patentable over Baecker et al. in view of Johnston, Jr. et al. Applicants further submit that claims 13, 29 and 31 are patentable over Baecker et al. in view of Johnston, Jr. et al., and that claim 7 is patentable over Baecker et al., for at least the reasons given with respect to claim 1.

The Examiner rejected claims 7, 19-21, 25 and 30 under 35 U.S.C. § 103(a) as being unpatentable over Johnston, Jr. et al. (OA, paragraph 6). With respect to claim 7, the Examiner alleged that Johnston, Jr. et al. discloses, *inter alia*, displaying state changes which correspond with state changes, detecting changing of a software state through the use of active animation, and cyclically displaying an animated sequence in the form of a rotating button. The Examiner

cited no reference as teaching the step of determining based on the detected event whether an animated sequence does not reflect the state of the container. The find button in Johnston, Jr. et al. is relied on by the Examiner as showing an animated sequence that reflects the state of a container. There is nothing at all in Johnston, Jr. et al. that discloses or suggests determining whether the find button does not reflect the state of a container. The Examiner appears to have attempted to overcome this deficiency of Johnston, Jr. et al. by stating that it would have been obvious to provide such a determination because doing so enables the computer programmer to provide a user-friendly method for the software user to track software states. Applicants submit that the Examiner has not provided an adequate motivation to modify Johnston, Jr. et al. to include the determination. The find button is nothing more than a trigger for activating a function to be performed. One of ordinary skill in the art would not be motivated to determine whether this find button reflects the state of a software container. Accordingly, Johnston, Jr. et al. does not disclose or suggest determining based on the detected event whether an animated sequence does not reflect the state of the container.

For at least the foregoing reasons, Applicants submit that claim 7 is patentable over Johnston, Jr., et al.

The present invention recited in claim 19, for example, recites a process for reflecting activity of a software container that is closed, comprising: detecting activity of the closed container; and updating an animated sequence so as to reflect activity of the closed container.

Applicants respectfully submit that Johnston, Jr. et al. does not disclose or suggest this claimed combination of steps. Among other things, the reference does not disclose or suggest detecting activity of the closed container, and updating an animated sequence so as to reflect activity of the closed container.

The Examiner indicated that it would have been obvious to include these steps in Johnston, Jr. et al. Applicants respectfully disagree with that position. There is no teaching or suggestion in Johnston, Jr. et al. to modify its system so that activity of the closed container is detected, and an animated sequence of the closed container is updated so as to reflect activity. Johnston, Jr. et al. discloses detecting the activity of a menu item (i.e., a button), and that an animated sequence is updated as to reflect activity of a button. A button is in no way analogous to a closed container. Accordingly, there is no suggestion in Johnston, Jr. et al. as to why one of ordinary skill in the art would modify the system of Johnston, Jr. et al. so that activity of the closed container is detected, and an animated sequence of the closed container is updated so as to reflect activity.

For at least the foregoing reasons, Applicants submit that claim 19 is patentable over Johnston, Jr., et al. Applicants further submit that claim 25 is patentable over Johnston, Jr., et al., and that claim 27 is patentable over Baecker et al., STN Express, and Johnston, Jr. et al., for at least the reasons given with respect to claim 19.

The Examiner rejected claims 22-24 and 26 under 35 U.S.C. § 103(a) as being unpatentable over STN Express, Lagarde et al., and Johnston, Jr. et al. (OA, paragraph 7). The Examiner alleged that STN Express discloses, *inter alia*, providing an emulator for a PC which emulates actions of a mainframe allowing a user on the PC to view actions of the mainframe, providing a status bar on the PC screen with the word 'online' or 'offline' depending on whether the emulator is connected to the mainframe, displaying the word 'online' in green when the mainframe awaits input from the PC user, displaying the word 'online' in red when the user is instructed to wait for output from the mainframe after data is entered. The Examiner argued that STN Express' use of an emulator can be interpreted as a method where a first computer has acted

LAW OFFICES

FINNEGAN, HENDERSON,
FARABOW, CARRETT,
& DUNNER, L.L.P.
1300 I STREET, N. W.
WASHINGTON, DC 20005
202-408-4000

upon a software container in a second computer. The Examiner also alleged that Lagarde et al. discloses coupling a database on a different server computer with a user computer equipped with a browser where the user requests information from the database server, and accessing data through a network. The Examiner further took official notice that it is widely accepted in the art to provide animated icons on an Internet server wherein the user of a second computer system accessing the server can execute software stored on the server from the second computer system. The Examiner then asserted that it would have been obvious to detect if a second computer system has acted upon a container and to update an animated sequence to be displayed on the first computer system so as to reflect the actions of the second computer system. The Examiner never mentioned how Johnston et al. was being relied upon.

The present invention recited in claim 22, for example, recites a process for reflecting activity of a network-based software container associated with a first computer system, comprising: detecting if a second computer system has acted upon the container; and updating an animated sequence to be displayed on the first computer system so as to reflect the actions of the second computer system.

Applicants respectfully submit that STN Express, Lagarde et al., and Johnston, Jr. et al. do not disclose or suggest this claimed combination of steps. Among other things, the references do not disclose or suggest updating an animated sequence to be displayed on the first computer system so as to reflect the actions of the second computer system.

The Examiner indicated that it would have been obvious to include this step, because doing so is a widely accepted method for allowing the user accessing a first computer system from a second computer system to view a copy of the output from the first computer from the second computer. Applicants respectfully disagree with that position. There is no teaching or

suggestion in STN Express, Lagarde et al., or Johnston, Jr. et al. to modify their systems so that an animated sequence to be displayed on the first computer system is updated so as to reflect the actions of the second computer system. None of the references actually show an animated sequence that is updated in the aforementioned manner. Instead, the Examiner took official notice and seems to be asserting that the missing features are obvious because they are well-known.

In response to previous requests to provide a reference, the Examiner supplied Gish (U.S. Patent No. 5,848,246) and maintained the rejection of claim 22, stating that the Official Notice can be replaced by Gish. The Examiner specifically argued that Gish provides animated icons on an Internet server wherein the user of a second computer system accessing the server can execute software stored on the server from the second computer system. Assuming, *arguendo*, that Gish provides animated icons as alleged by the Examiner, Applicants fail to comprehend how Gish's teaching of an animated icon renders it obvious to modify the system of STN Express, Lagarde et al., or Johnston, Jr. et al., so that an animated sequence to be displayed on the first computer system is updated so as to reflect the actions of the second computer system. Applicants submit that the animated icon disclosed in Gish is not sufficient to disclose or suggest updating an animated sequence to be displayed on the first computer system so as to reflect the actions of the second computer system.

For at least the foregoing reasons, Applicants submit that claim 22 is patentable over STN Express, Lagarde et al., and Johnston, Jr. et al. Applicants further submit that claim 26 is patentable over STN Express, Lagarde et al., and Johnston, Jr. et al., and that claim 28 is patentable over Baecker et al., STN Express, and Johnston, Jr. et al., for at least the reasons given with respect to claim 22.

LAW OFFICES

FINNEGAN, HENDERSON,
FARABOW, GARRETT,
& DUNNER, L.L.P.
1300 I STREET, N.W.
WASHINGTON, DC 20005
202-408-4000

New claim 32 recites a process for reflecting a state of a software container having objects, comprising: cyclically displaying a series of frames reflecting a state of the container as an animated sequence; detecting an event reflecting a change in the state of the container, wherein the container is a web page related to user discussion; determining based on the detected event whether an animated sequence does not reflect the state of the container; and updating the cyclical display based on the determination.

Applicants respectfully submit that none of the previously cited references disclose or suggest a process for reflecting a state of a software container in which the software container is a web page related to user discussion. Accordingly, new claim 32 is patentable over the cited references.

The dependent claims are allowable for at least the reasons stated above with regard to their respective base claims.

Since each of the independent claims have been placed in allowable form, Applicants respectfully request the timely allowance of this application.

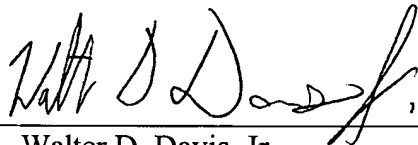
LAW OFFICES

FINNEGAN, HENDERSON,
FARABOW, CARRETT,
& DUNNER, L.L.P.
1300 I STREET, N. W.
WASHINGTON, DC 20005
202-408-4000

If an extension of time under 37 C.F.R. § 1.136 is required to obtain entry of this response, such extension is hereby requested. If there are any fees due under 37 C.F.R. § 1.16 or 1.17 which are not enclosed, including any fees required for an extension of time under 37 C.F.R. § 1.136, please charge those fees to our Deposit Account No. 06-0916.

Respectfully submitted,

FINNEGAN, HENDERSON, FARABOW,
GARRETT & DUNNER, L.L.P.

By: 
Walter D. Davis, Jr.
Reg. No. 45,137

Dated: October 12, 2000